


```
BBBBBBBBB      AAAAAA      SSSSSSSSS      SSSSSSSSS      YY      YY      SSSSSSSSS
BBBBBBBBB      AAAAAA      SSSSSSSSS      SSSSSSSSS      YY      YY      SSSSSSSSS
BB      BB      AA      AA      SS      SS      YY      YY      SS      SS
BB      BB      AA      AA      SS      SS      YY      YY      SS      SS
BB      BB      AA      AA      SS      SS      YY      YY      SS      SS
BBBBBBBBB      AA      AA      SSSSSSS      SSSSSSS      YY      YY      SSSSSSS
BBBBBBBBB      AA      AA      SSSSSSS      SSSSSSS      YY      YY      SSSSSSS
BB      BB      AAAAAAAAAA      SS      SS      YY      YY      SS      SS
BB      BB      AAAAAAAAAA      SS      SS      YY      YY      SS      SS
BB      BB      AA      AA      SS      SS      YY      YY      SS      SS
BB      BB      AA      AA      SSSSSSS      SSSSSSS      YY      SSSSSSS
BBBBBBBBB      AA      AA      SSSSSSS      SSSSSSS      YY      SSSSSSS
BBBBBBBBB      AA      AA      SSSSSSS      SSSSSSS      YY      SSSSSSS
```

.....
.....
.....
.....

```
LL      IIIIII      SSSSSSSSS
LL      IIIIII      SSSSSSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SSSSSSS
LL      II      SSSSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SS
LLLLLLLLLL      IIIIII      SSSSSSSSS
LLLLLLLLLL      IIIIII      SSSSSSSSS
```



```
1 0001 0 MODULE BAS$$SYS (
2 0002 0 IDENT = '1-014'
3 0003 0 ) =
4 0004 1 BEGIN
5 0005 1
6 0006 1 *****
7 0007 1 *
8 0008 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
9 0009 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
10 0010 1 * ALL RIGHTS RESERVED.
11 0011 1 *
12 0012 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
13 0013 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
14 0014 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
15 0015 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
16 0016 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
17 0017 1 * TRANSFERRED.
18 0018 1 *
19 0019 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
20 0020 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
21 0021 1 * CORPORATION.
22 0022 1 *
23 0023 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
24 0024 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
25 0025 1 *
26 0026 1 *
27 0027 1 *****
28 0028 1
29 0029 1
30 0030 1 ++
31 0031 1 FACILITY: VAX-11 BASIC RSTS COMPATABILITY
32 0032 1
33 0033 1 ABSTRACT:
34 0034 1
35 0035 1 This module contains the RSTS-compatible SYS function.
36 0036 1 Some of the more complex functions do calls to BPA routines.
37 0037 1
38 0038 1 ENVIRONMENT: VAX-11 User Mode
39 0039 1
40 0040 1 AUTHOR: John Sauter, CREATION DATE: 01-OCT-1979
41 0041 1
42 0042 1 MODIFIED BY:
43 0043 1
44 0044 1 1-001 - Original.
45 0045 1 1-002 - Define BPAS_HISEG as 0, to disable the checking for the end
46 0046 1 of the compatibility-mode high segment, and allocate
47 0047 1 some global cells for ASSIGN/DEASSIGN. JBS 02-OCT-1979
48 0048 1 1-003 - Instead of passing the XRB to the message send/receive code,
49 0049 1 pass the relevant fields of the XRB. Also, BPAS_HISEG is
50 0050 1 no longer needed. JBS 04-OCT-1979
51 0051 1 1-004 - Debug message send/receive. JBS 05-OCT-1979
52 0052 1 1-005 - Continue debugging message send/receive. JBS 07-OCT-1979
53 0053 1 1-006 - Fix some errors in calling terminal functions. JBS 12-OCT-1979
54 0054 1 1-007 - Handle short parameter strings correctly. JBS 17-OCT-1979
55 0055 1 1-008 - Add core common. JBS 03-DEC-1979
56 0056 1 1-009 - Add FSS. JBS 04-DEC-1979
57 0057 1 1-010 - Add setting priority. JBS 04-DEC-1979
```

```
! RSTS SYS Function
! File: BASSYS.B32 Edit: PL1014
```

BASSSYS
1-014

K 3
16-Sep-1984 01:16:51 VAX-11 Bliss-32 V4.0-742
14-Sep-1984 11:56:41 [BASRTL.SRC]BASSYS.B32;1

Page 2
(1)

```
: 58      0058 1 | 1-011 - Make "exit with no prompt" exit more quietly, fix a typo in putting
: 59      0059 1 |      into core common, and make ODI submode give an error message, since
: 60      0060 1 |      there isn't time to make it work right for this release. JBS 26-FEB-1980
: 61      0061 1 | 1-012 - Put in small send and receive. All four functions. FM 24-FEB-81.
: 62      0062 1 | 1-013 - LIB$STOP should be declared EXTERNAL. PL 20-Nov-81
: 63      0063 1 | 1-014 - Call BASS$STOP to signal errors instead of BASS$STOP_IO. PLL 16-Jun-1982
: 64      0064 1 | --
: 65      0065 1 |
: 66      0066 1 | <BLF/PAGE>
```



```
68 0067 1 |
69 0068 1 | SWITCHES:
70 0069 1 |
71 0070 1 |
72 0071 1 | SWITCHES ADDRESSING_MODE (EXTERNAL = GENERAL, NONEXTERNAL = WORD_RELATIVE);
73 0072 1 |
74 0073 1 |
75 0074 1 | LINKAGES:
76 0075 1 |
77 0076 1 |
78 0077 1 | REQUIRE 'RTLIN:OTSLNK';           ! I/O Linkages
79 0506 1 |
80 0507 1 |
81 0508 1 | TABLE OF CONTENTS:
82 0509 1 |
83 0510 1 |
84 0511 1 | FORWARD ROUTINE
85 0512 1 |     BASSSYS : NOVALUE,           ! Do a SYS() function
86 0513 1 |     BASS$UUD : NOVALUE;         ! Do a UUD sub-function
87 0514 1 |
88 0515 1 |
89 0516 1 | INCLUDE FILES:
90 0517 1 |
91 0518 1 |
92 0519 1 | REQUIRE 'RTLML:OTSLUB';         ! LUB definitions
93 0659 1 |
94 0660 1 | REQUIRE 'RTLIN:BPASTRUCT';     ! Structuring definitions
95 0751 1 |
96 0752 1 | REQUIRE 'RTLIN:BPAFQBDEF';     ! FIRQB definitions
97 0876 1 |
98 0877 1 | REQUIRE 'RTLIN:BPAFSBDEF';     ! FSB definitions
99 1008 1 |
100 1009 1 | REQUIRE 'RTLIN:BPAFUNDEF';     ! Function definitions
101 1259 1 |
102 1260 1 | REQUIRE 'RTLIN:RTLPSECT';      ! Macros for defining psects
103 1355 1 |
104 1356 1 | LIBRARY 'RTLSTARLE';          ! System symbols
105 1357 1 |
106 1358 1 |
107 1359 1 | MACROS:
108 1360 1 |
109 1361 1 |     NONE
110 1362 1 |
111 1363 1 | EQUATED SYMBOLS:
112 1364 1 |
113 1365 1 |     NONE
114 1366 1 |
115 1367 1 | PSECTS:
116 1368 1 |
117 1369 1 | DECLARE_PSECTS (BAS);         ! Declare psects for BASS$ facility
118 1370 1 |
119 1371 1 | OWN STORAGE:
120 1372 1 |
121 1373 1 |     NONE
122 1374 1 |
123 1375 1 | EXTERNAL REFERENCES:
124 1376 1 |
```

```
125 1377 1
126 1378 1 EXTERNAL ROUTINE
127 1379 1   LIB$STOP : NOVALUE,
128 1380 1   STR$COPY_DX,
129 1381 1   STR$COPY_R,
130 1382 1   STR$FREE_DX,
131 1383 1   STR$CONCAT,
132 1384 1   LIB$GET_COMMON,
133 1385 1   LIB$PUT_COMMON,
134 1386 1   BASS$STOP : NOVALUE,
135 1387 1   BASS$CTRL0 : NOVALUE,
136 1388 1   BASS$NOECHO : NOVALUE,
137 1389 1   BASS$ONECHR : NOVALUE,
138 1390 1   BASS$CANTYPAHEAD : NOVALUE,
139 1391 1   BASS$CTRLC : NOVALUE,
140 1392 1   BASS$ECHO : NOVALUE,
141 1393 1   BASS$ERT : NOVALUE,
142 1394 1   BASS$CB_PUSH : JSB CB PUSH NOVALUE,
143 1395 1   BASS$CB_POP : JSB CB POP NOVALUE,
144 1396 1   BASS$STOP_IO : NOVALUE,
145 1397 1   BPAS$MESAG,
146 1398 1   BPAS$ASSIGN,
147 1399 1   BPAS$DEASSIGN,
148 1400 1   BPAS$DEASS ALL,
149 1401 1   BPAS$SET_DEF,
150 1402 1   BPAS$FSS,
151 1403 1   BPAS$FREE BLOCK,
152 1404 1   BPAS$SET_PRI;
153 1405 1
154 1406 1 !+
155 1407 1 ! The following are the error codes used in this module.
156 1408 1 !-
157 1409 1
158 1410 1 EXTERNAL LITERAL
159 1411 1   BASS$K_ILLIO_CHA : UNSIGNED (8),
160 1412 1   BASS$K_IO_CHANOT : UNSIGNED (8),
161 1413 1   BASS$K_ILSYSUSA : UNSIGNED (8),
162 1414 1   BASS$K_ILLBYTCOU : UNSIGNED (8),
163 1415 1   BASS$K_NO_ROOMUSE : UNSIGNED (8),
164 1416 1   BASS$K_MISSPEFEA : UNSIGNED (8);
165 1417 1
```

! Signal a fatal error
! Copy a string by descriptor
! Copy a string by reference
! Free a string
! Concatenate strings
! Fetch from process common
! Store in process common
! Signal fatal error
! Clear control 0
! Turn off echoing
! Enter ODT submode
! Cancel type-ahead
! Enable control-C trapping
! Turn of echoing
! Return error message
! Load register CCB
! Done with register CCB
! Signal fatal I/O error
! Do send/receive
! Assign a device
! Deassign a device
! Deassign all devices
! Set default
! File string scan
! Free heap storage
! Change priority

! Illegal I/O channel
! I/O channel not open
! Illegal SYSS() usage
! Illegal byte count for I/O
! No room for user on device
! Missing special feature


```
167 1418 1 GLOBAL ROUTINE BAS$$SYS (
168 1419 1     RESULT_STR,
169 1420 1     CODE_STR
170 1421 1 ) : NOVALUE =
171 1422 1
172 1423 1 ++
173 1424 1 FUNCTIONAL DESCRIPTION:
174 1425 1
175 1426 1     Do a RSTS/E compatible SYS() function.
176 1427 1
177 1428 1 FORMAL PARAMETERS:
178 1429 1
179 1430 1     RESULT_STR.wz.dx      Result of the SYS() function
180 1431 1     CODE_STR.rz.dx        A string of bytes which specify what to do.
181 1432 1
182 1433 1 IMPLICIT INPUTS:
183 1434 1
184 1435 1     NONE
185 1436 1
186 1437 1 IMPLICIT OUTPUTS:
187 1438 1
188 1439 1     NONE
189 1440 1
190 1441 1 ROUTINE VALUE:
191 1442 1
192 1443 1     Depends on the function, see above. Where the result is not
193 1444 1     defined the null string is returned.
194 1445 1
195 1446 1 SIDE EFFECTS:
196 1447 1
197 1448 1     Signals if an error is encountered.
198 1449 1
199 1450 1 --
200 1451 1
201 1452 2 BEGIN
202 1453 2
203 1454 2 MAP
204 1455 2     CODE_STR : REF BLOCK [8, BYTE];
205 1456 2
206 1457 2 BIND
207 1458 2     FIRST_BYTE = CODE_STR [DSC$A_POINTER] : REF VECTOR [, BYTE],
208 1459 2     STR_LENGTH = .CODE_STR [DSC$W_LENGTH];
209 1460 2
210 1461 2 LOCAL
211 1462 2     RET_STRING : BLOCK [8, BYTE];
212 1463 2
213 1464 2     RET_STRING [DSC$W_LENGTH] = 0;
214 1465 2     RET_STRING [DSC$B_DTYPE] = DSC$K_DTYPE_Z;
215 1466 2     RET_STRING [DSC$B_CLASS] = DSC$K_CLASS_D;
216 1467 2     RET_STRING [DSC$A_POINTER] = 0;
217 1468 2
218 1469 2 ++ Dispatch on the first byte of the string.
219 1470 2 --
220 1471 2
221 1472 3 IF (STR_LENGTH EQLU 0)
222 1473 2 THEN
223 1474 2     BAS$$STOP (BAS$K_ILLSYSUSA)
```



```

224 1475 2 ELSE
225 1476
226 1477 CASE .FIRST_BYTE [0] FROM 0 TO 12 OF
227 1478 SET
228 1479
229 1480 [0] : ! Cancel control 0
230 1481 BASSRCTRLO ((IF (STR_LENGTH GEQ 2) THEN .FIRST_BYTE [1] ELSE 0));
231 1482
232 1483 [1] : ! Enter TAPE mode
233 1484 BASS$STOP (BASS$K_MISSPEFEA);
234 1485
235 1486 [2] : ! Enable echoing
236 1487 BAS$ECHO ((IF (STR_LENGTH GEQ 2) THEN .FIRST_BYTE [1] ELSE 0));
237 1488
238 1489 [3] : ! Disable echoing
239 1490 BAS$NOECHO ((IF (STR_LENGTH GEQ 2) THEN .FIRST_BYTE [1] ELSE 0));
240 1491
241 1492 [4] : ! ODT submode
242 1493 BEGIN
243 1494 BASS$STOP (BASS$K_MISSPEFEA);
244 1495 BAS$ONECHR ((IF (STR_LENGTH GEQ 2) THEN .FIRST_BYTE [1] ELSE 0));
245 1496 END;
246 1497
247 1498 [5] : ! Exit with no prompt
248 1499 $EXIT (CODE = SS$_NORMAL);
249 1500
250 1501 [6] : ! Call file processor
251 1502 BAS$UUD (RET_STRING, .CODE_STR);
252 1503
253 1504 [7] : ! Get core common
254 1505 BEGIN
255 1506 LOCAL
256 1507 STATUS;
257 1508
258 1509 STATUS = LIB$GET_COMMON (RET_STRING);
259 1510
260 1511 IF ( NOT .STATUS) THEN LIB$STOP (.STATUS);
261 1512
262 1513 END;
263 1514
264 1515 [8] : ! Put core common
265 1516 BEGIN
266 1517 LOCAL
267 1518 STATUS,
268 1519 LEN,
269 1520 DESC : BLOCK [8, BYTE];
270 1521
271 1522 LEN = STR_LENGTH - 1;
272 1523
273 1524 IF (.LEN GTR 127) THEN LEN = 0;
274 1525
275 1526 DESC [DSC$_LENGTH] = .LEN;
276 1527 DESC [DSC$_DTYPE] = DSC$K_DTYPE_Z;
277 1528 DESC [DSC$_CLASS] = DSC$K_CLASS_S;
278 1529 DESC [DSC$_POINTER] = FIRST_BYTE [1];
279 1530
280 1531 3
```



```
: 281 1532 3
: 282 1533 3
: 283 1534 3
: 284 1535 3
: 285 1536 3
: 286 1537 3
: 287 1538 3
: 288 1539 3
: 289 1540 3
: 290 1541 3
: 291 1542 3
: 292 1543 3
: 293 1544 3
: 294 1545 3
: 295 1546 3
: 296 1547 3
: 297 1548 3
: 298 1549 3
: 299 1550 3
: 300 1551 3
: 301 1552 3
: 302 1553 3
: 303 1554 3
: 304 1555 3
: 305 1556 3
: 306 1557 1
```

```
STATUS = LIB$PUT_COMMON (DESC);
IF ( NOT .STATUS) THEN LIB$STOP (.STATUS);
END;
[9] : ! Exit and clear program
$EXIT (CODE = SSS_NORMAL);
[10] : ! Special implementation
BAS$$STOP (BAS$K_MISSPEFEA);
[11] : ! Cancel type ahead
BAS$CANTYPAHEAD ((IF (STR_LENGTH GEQ 2) THEN .FIRST_BYTE [1] ELSE 0));
[12] : ! Info on last opened file
BAS$$STOP (BAS$K_MISSPEFEA);
[OUTRANGE] : ! Not defined
BAS$$STOP (BAS$K_ILLSYSUSA);
TES;
STR$COPY_DX (.RESULT_STR, RET_STRING);
STR$FREE_DX (RET_STRING);
RETURN;
END; ! of routine BAS$SYS
```

```
.TITLE BAS$SYS
.IDENT \1-014\
```

```
.EXTRN LIB$STOP, STR$COPY_DX
.EXTRN STR$COPY_R, STR$FREE1_DX
.EXTRN STR$CONCAT, LIB$GET_COMMON
.EXTRN LIB$PUT_COMMON, BAS$$STOP
.EXTRN BAS$RCTRL, BAS$NOECHO
.EXTRN BAS$ONECHR, BAS$CANTYPAHEAD
.EXTRN BAS$CTRLC, BAS$ECHO
.EXTRN BAS$ERT, BAS$$CB_PUSH
.EXTRN BAS$$CB_POP, BAS$$STOP_IO
.EXTRN BPAS$MSG, BPAS$ASSIGN
.EXTRN BPAS$DEASSIGN, BPAS$DEASS_ALL
.EXTRN BPAS$SET_DEF, BPAS$FSS
.EXTRN BPAS$FREE_BLOCK, BPAS$SET_PRI
.EXTRN BAS$K_ILC10_CHA
.EXTRN BAS$K_IO_CHANOT
.EXTRN BAS$K_ILSYSUSA
.EXTRN BAS$K_ILBYTCOU
.EXTRN BAS$K_NO_RDOUSE
.EXTRN BAS$K_MISSPEFEA
.EXTRN SYS$EXIT
```

```
.PSECT _BAS$CODE, NOWRT, SHR, PIC, 2
```

```
.ENTRY BAS$SYS, Save R2,R3,R4
MOVAB BAS$$STOP, R4
SUBL2 #16, SP
```

```
001C 00000
54 00000000G 00 9E 00002
5E 10 C2 00009
```

```
: 1418
:
:
```

50	08	AC	04	C1	0000C	ADDL3	#4, CODE STR, R0	1458
		53	BC	3C	00011	MOVZWL	@CODE STR, R3	1459
	08	AE	8F	D0	00015	MOVL	#3355432, RET_STRING	1464
			AE	D4	0001D	CLRL	RET_STRING+4	1467
			53	D5	00020	TSTL	R3	1472
			21	13	00022	BEQL	2\$	
		52	60	D0	00024	MOVL	(R0), R2	1477
	OC	00	62	8F	00027	CASEB	(R2), #0, #12	
004D	0037	00EE	0021		0002B	.WORD	3\$-1\$,-	
008D	0080	00CD	0063		00033		27\$-1\$,-	
00D8	00EE	00CD	0099		0003B		6\$-1\$,-	
			00EE		00043		9\$-1\$,-	
							12\$-1\$,-	
							22\$-1\$,-	
							15\$-1\$,-	
							17\$-1\$,-	
							18\$-1\$,-	
							22\$-1\$,-	
							27\$-1\$,-	
							24\$-1\$,-	
							27\$-1\$	
	7E	00G	8F	9A	00045	MOVZBL	#BAS\$K_ILLSYSUSA, -(SP)	1551
			00D1	31	00049	BRW	28\$	
	02		53	B1	0004C	CMPW	R3, #2	1481
			06	1F	0004F	BLSSU	4\$	
	7E	01	A2	9A	00051	MOVZBL	1(R2), -(SP)	
			02	11	00055	BRB	5\$	
			7E	D4	00057	CLRL	-(SP)	
00000000G	00		01	FB	00059	CALLS	#1, BAS\$RCTRL0	
			54	11	00060	BRB	16\$	
	02		53	B1	00062	CMPW	R3, #2	1487
			06	1F	00065	BLSSU	7\$	
	7E	01	A2	9A	00067	MOVZBL	1(R2), -(SP)	
			02	11	0006B	BRB	8\$	
00000000G	00		7E	D4	0006D	CLRL	-(SP)	
			01	FB	0006F	CALLS	#1, BAS\$ECHO	
			7E	11	00076	BRB	21\$	
	02		53	B1	00078	CMPW	R3, #2	1490
			06	1F	0007B	BLSSU	10\$	
	7E	01	A2	9A	0007D	MOVZBL	1(R2), -(SP)	
			02	11	00081	BRB	11\$	
00000000G	00		7E	D4	00083	CLRL	-(SP)	
			01	FB	00085	CALLS	#1, BAS\$NOECHO	
			73	11	0008C	BRB	23\$	
	7E	00G	8F	9A	0008E	MOVZBL	#BAS\$K_MISSPEFEA, -(SP)	1494
64			01	FB	00092	CALLS	#1, BAS\$\$STOP	
02			53	B1	00095	CMPW	R3, #2	1495
			06	1F	00098	BLSSU	13\$	
	7E	01	A2	9A	0009A	MOVZBL	1(R2), -(SP)	
			02	11	0009E	BRB	14\$	
00000000G	00		7E	D4	000A0	CLRL	-(SP)	
			01	FB	000A2	CALLS	#1, BAS\$ONECHR	
			75	11	000A9	BRB	29\$	1477
		08	AC	DD	000AB	PUSHL	CODE STR	1502
		OC	AE	9F	000AE	PUSHAB	RET_STRING	
			02	FB	000B1	CALLS	#2, BAS\$\$UUO	
0000V	CF		68	11	000B6	BRB	29\$	

00000000G	00	08	AE	9F	000B8	17\$:	PUSHAB	RET_STRING	: 1510
			01	FB	000BB		CALLS	#1, LIB\$GET_COMMON	: 1512
	50		26	11	000C2		BRB	20\$: 1524
0000007F	8F	FF	A3	9E	000C4	18\$:	MOVAB	-1(R3), LEN	: 1526
			50	D1	000C8		CMPL	LEN, #127	: 1528
			02	15	000CF		BLEQ	19\$: 1529
	6E		50	D4	000D1		CLRL	LEN	: 1531
	02	0100	50	B0	000D3	19\$:	MOVW	LEN, DESC	: 1532
	04	01	8F	B0	000D6		MOVW	#256, DESC+2	: 1534
			A2	9E	000DC		MOVAB	1(R2), DESC+4	: 1539
00000000G	00		5E	DD	000E1		PUSHL	SP	: 1545
	33		01	FB	000E3		CALLS	#1, LIB\$PUT_COMMON	: 1547
			50	E8	000EA	20\$:	BLBS	STATUS, 29\$: 1548
00000000G	00		50	DD	000ED		PUSHL	STATUS	: 1549
			01	FB	000EF		CALLS	#1, LIB\$STOP	: 1550
			28	11	000F6	21\$:	BRB	29\$: 1551
00000000G	00		01	DD	000F8	22\$:	PUSHL	#1	: 1552
			01	FB	000FA		CALLS	#1, SYS\$EXIT	: 1553
	02		1D	11	00101	23\$:	BRB	29\$: 1554
			53	B1	00103	24\$:	CMPL	R3, #2	: 1555
	7E	01	06	1F	00106		BLSSU	25\$: 1556
			A2	9A	00108		MOVZBL	1(R2), -(SP)	: 1557
			02	11	0010C		BRB	26\$: 1558
00000000G	00		7E	D4	0010E	25\$:	CLRL	-(SP)	: 1559
			01	FB	00110	26\$:	CALLS	#1, BASS\$CANTYPAHEAD	: 1560
	7E	00G	07	11	00117		BRB	29\$: 1561
	64		8F	9A	00119	27\$:	MOVZBL	#BASS\$K, MISSPEFEA, -(SP)	: 1562
			01	FB	0011D	28\$:	CALLS	#1, BASS\$STOP	: 1563
		08	AE	9F	00120	29\$:	PUSHAB	RET_STRING	: 1564
		04	AC	DD	00123		PUSHL	RESULT_STR	: 1565
00000000G	00		02	FB	00126		CALLS	#2, STR\$COPY_DX	: 1566
		08	AE	9F	0012D		PUSHAB	RET_STRING	: 1567
00000000G	00		01	FB	00130		CALLS	#1, STR\$FREE1_DX	: 1568
			04	00137		RET			: 1569

; Routine Size: 312 bytes, Routine Base: _BASS\$CODE + 0000

; 307 1558 1

```

: 309      1559 1 ROUTINE BASS$UUD (
: 310      1560 1     RESULT_STR,
: 311      1561 1     CODE_STR
: 312      1562 1     ) : NOVALUE =
: 313      1563 1
: 314      1564 1
: 315      1565 1
: 316      1566 1
: 317      1567 1
: 318      1568 1
: 319      1569 1
: 320      1570 1
: 321      1571 1
: 322      1572 1
: 323      1573 1
: 324      1574 1
: 325      1575 1
: 326      1576 1
: 327      1577 1
: 328      1578 1
: 329      1579 1
: 330      1580 1
: 331      1581 1
: 332      1582 1
: 333      1583 1
: 334      1584 1
: 335      1585 1
: 336      1586 1
: 337      1587 1
: 338      1588 1
: 339      1589 1
: 340      1590 1
: 341      1591 1
: 342      1592 1
: 343      1593 1
: 344      1594 2
: 345      1595 2
: 346      1596 2
: 347      1597 2
: 348      1598 2
: 349      1599 2
: 350      1600 2
: 351      1601 2
: 352      1602 2
: 353      1603 2
: 354      1604 2
: 355      1605 2
: 356      1606 2
: 357      1607 2
: 358      1608 2
: 359      1609 2
: 360      1610 2
: 361      1611 2
: 362      1612 2
: 363      1613 2
: 364      1614 2
: 365      1615 2

ROUTINE BASS$UUD (
    RESULT_STR,
    CODE_STR
) : NOVALUE =

    ++
    FUNCTIONAL DESCRIPTION:
        Do a RSTS/E compatible SYS() function, where the first byte of
        the string is a 6.

    FORMAL PARAMETERS:
        RESULT_STR.wz.dx    The result of the SYS() function
        CODE_STR.rz.dx      A string of bytes which specify what to do.

    IMPLICIT INPUTS:
        NONE

    IMPLICIT OUTPUTS:
        NONE

    ROUTINE VALUE:
        Depends on the function, see above. Where the result is not
        specified, the user's string is set to null.

    SIDE EFFECTS:
        Signals if an error is encountered.

    --

BEGIN
EXTERNAL ROUTINE
    R5OASC : NOVALUE;

MAP
    CODE_STR : REF BLOCK [8, BYTE];

LOCAL
    RET_STRING : BLOCK [8, BYTE],
    FIRQB : BLOCK [512 + 41 + 2, BYTE] FIELD (FQB$FIELDS),
    FIRQB_DESC : BLOCK [8, BYTE];

    FIRQB_DESC [DSC$W_LENGTH] = 512 + 41;
    FIRQB_DESC [DSC$B_DTYPE] = DSC$K_DTYPE_BU;
    FIRQB_DESC [DSC$B_CLASS] = DSC$K_CLASS_S;
    FIRQB_DESC [DSC$A_POINTER] = FIRQB [2, B_];
    CH$COPY (.CODE_STR [DSC$W_LENGTH], .CODE_STR [DSC$A_POINTER], 0, 512 + 41, FIRQB [2, B_]);
    RET_STRING [DSC$W_LENGTH] = 0;
    RET_STRING [DSC$B_DTYPE] = DSC$K_DTYPE_Z;
    RET_STRING [DSC$B_CLASS] = DSC$K_CLASS_D;
    RET_STRING [DSC$A_POINTER] = 0;

```



```

366 1616 2 1+ Dispatch on the second byte of the string.
367 1617 2 1-
368 1618 2
369 1619 2
370 1620 2 CASE .FIRQB [FQBSB_FUNCTION] FROM FUN$K_MINUUD TO FUN$K_MAXUUD OF
371 1621 2 SET
372 1622 2
373 1623 2 [FUN$K_UUCCT] : ! Control C trap enable
374 1624 2 BASS$CTRLC ();
375 1625 2
376 1626 2 [FUN$K_UUERR] : ! Get error message
377 1627 2 BEGIN
378 1628 2
379 1629 2 LOCAL
380 1630 2 ERR_STRING : BLOCK [8, BYTE],
381 1631 2 HEADER : BLOCK [8, BYTE];
382 1632 2
383 1633 2 ERR_STRING [DSC$W_LENGTH] = 0;
384 1634 2 ERR_STRING [DSC$B_DTYPE] = DSC$K_DTYPE_T;
385 1635 2 ERR_STRING [DSC$B_CLASS] = DSC$K_CLASS_D;
386 1636 2 ERR_STRING [DSC$A_POINTER] = 0;
387 1637 2 BASS$RT (ERR_STRING, .FIRQB [FQBSB_ERRNUM]);
388 1638 2 HEADER [DSC$W_LENGTH] = 2;
389 1639 2 HEADER [DSC$B_DTYPE] = DSC$K_DTYPE_BU;
390 1640 2 HEADER [DSC$B_CLASS] = DSC$K_CLASS_S;
391 1641 2 HEADER [DSC$A_POINTER] = UPLIT (BYTE (0, 0));
392 1642 2 STR$CONCAT (RET_STRING, HEADER, ERR_STRING);
393 1643 2 STR$FREE1_DX (ERR_STRING);
394 1644 2 END;
395 1645 2
396 1646 2 [FUN$K_UUMES1] : ! Small message send/receive
397 1647 2 BEGIN
398 1648 2
399 1649 2 LITERAL
400 1650 2 K_SML_SEND = -1,
401 1651 2 K_SML_REMREC = 0,
402 1652 2 K_SML_DCLREC_REC = 1,
403 1653 2 K_SML_REC = 2;
404 1654 2
405 1655 2 LOCAL
406 1656 2 BYTXFR, ! Number of bytes actually transferred
407 1657 2 ASCII_LOGNAM : VECTOR [6, BYTE], ! A buffer to put the translated R50 name.
408 1658 2 RES_STRING : VECTOR [30, BYTE], ! Some place to put the result string temporarily.
409 1659 2 MESAG : VECTOR [20, BYTE]; ! A temp. place to put the message.
410 1660 2
411 1661 2 1+ Translate the name passed to ASCII.
412 1662 2 1-
413 1663 2
414 1664 2 R50ASC (%REF (6), FIRQB [FQBS$RCVNAM], ASCII_LOGNAM);
415 1665 2
416 1666 2 IF .ASCII_LOGNAM [0] EQL %C'?' THEN BASS$$STOP_IO (BASS$K_ILLSYSUSA);
417 1667 2
418 1668 2 1+ Do each function seperately
419 1669 2 1-
420 1670 2
421 1671 2 CASE .FIRQB [FQBSB_SUBFUN] FROM K_SML_SEND TO K_SML_REC OF
422 1672 2
```

```

: 423      1673      3
: 424      1674      3
: 425      1675      3
: 426      1676      4
: 427      1677      4
: 428      1678      4
: 429      1679      4
: 430      1680      3
: 431      1681      3
: 432      1682      3
: 433      1683      3
: 434      1684      3
: 435      1685      3
: 436      1686      4
: 437      1687      4
: 438      1688      4
: 439      1689      4
: 440      1690      4
: 441      1691      4
: 442      1692      4
: 443      1693      3
: 444      1694      3
: 445      1695      3
: 446      1696      4
: 447      1697      4
: 448      1698      4
: 449      1699      4
: 450      1700      4
: 451      1701      4
: 452      1702      4
: 453      1703      4
: 454      1704      4
: 455      1705      3
: 456      1706      3
: 457      1707      3
: 458      1708      3
: 459      1709      3
: 460      1710      3
: 461      1711      3
: 462      1712      3
: 463      1713      3
: 464      1714      3
: 465      1715      3
: 466      1716      3
: 467      1717      3
: 468      1718      3
: 469      1719      3
: 470      1720      3
: 471      1721      3
: 472      1722      3
: 473      1723      3
: 474      1724      3
: 475      1725      3
: 476      1726      3
: 477      1727      3
: 478      1728      3
: 479      1729      3

SET
[K_SML_SEND] :
    BEGIN
    CH$MOVE (20, FIRQB [10, B_], MESAG);
    CH$MOVE (20, MESAG, FIRQB [12, B_]);           !The message
    CH$MOVE (6, ASCII_LOGNAM, FIRQB [FQB$T_RCVNAM]); !Logical name
    END;

[K_SML_REMREC] :
    1;                                           ! Looks the same, so don't do anything.

[K_SML_DCLREC_REC] :
    BEGIN
    CH$MOVE (6, ASCII_LOGNAM, FIRQB [FQB$T_RCVNAM]); !Logical name
    FIRQB [FQB$W_BMAX] = -1;           ! Use temporary mailboxes.
    FIRQB [FQB$B_ACCESS] = 1;          ! Local message.
    BP$MESAG (FIRQB, 0, 0, BYTXFR);
    CH$FILL (0, 32, FIRQB [2, B_]);
    FIRQB [FQB$B_SUBFUN] = K_SML_REC;
    END;

[K_SML_REC] :
    BEGIN
    CH$MOVE (6, ASCII_LOGNAM, FIRQB [FQB$T_RCVNAM]); !Logical name
    FIRQB [FQB$B_SUBFUN] = K_SML_DCLREC_REC;
    FIRQB [FQB$W_BMAX] = -1;           ! Use temporary mailboxes.
    FIRQB [FQB$B_ACCESS] = 1;          ! Local message.
    BP$MESAG (FIRQB, 0, 0, BYTXFR);
    CH$FILL (0, 32, FIRQB [2, B_]);
    FIRQB [FQB$B_SUBFUN] = K_SML_REC;
    FIRQB [FQB$B_RMOD] = 1;           !Sleep indefinitely
    END;

TES;

!+
!- Now call BP$MESAG, to do the work.
    BP$MESAG (FIRQB, 0, 0, BYTXFR);
    CH$FILL (0, 30, RES_STRING);
    CH$MOVE (20, FIRQB [FQB$T_PAR_STR], RES_STRING + 8);
    STR$COPY_R (RET_STRING, %REF (30), RES_STRING);
    END;

[FUN$K_UUMES2] :
    BEGIN
    GLOBAL REGISTER
    CCB = K_CCB_REG : REF BLOCK [, BYTE];

    LOCAL
    BUFLN,
    BUFADR,
    BYTXFR;
    ! Length of buffer
    ! Address of user's buffer
    ! Number of bytes actually transferred

!+
!- Set up buffer length, byte count and buffer address based on the SYS()
```



```

: 480      1730 3  |_string.
: 481      1731 3  |_-
: 482      1732 3  |
: 483      1733 4      IF (.FIRQB [12, B_] EQL 0)
: 484      1734 3      THEN
: 485      1735 4          BEGIN
: 486      1736 4      |+_The buffer is in the string.
: 487      1737 4      |_-
: 488      1738 4          CCB = 0;
: 489      1739 4
: 490      1740 4          IF (.CODE_STR [DSC$W_LENGTH] GTR 40)
: 491      1741 5              THEN
: 492      1742 4                  BEGIN
: 493      1743 5                      BUFADR = FIRQB [42, B_];
: 494      1744 5                      BUFLN = .CODE_STR [DSC$W_LENGTH] - 40;
: 495      1745 5                      END
: 496      1746 5                  ELSE
: 497      1747 4                      BEGIN
: 498      1748 5
: 499      1749 5      |+_There is no buffer.
: 500      1750 5      |_-
: 501      1751 5                      BUFADR = BUFLN = 0;
: 502      1752 5                      END
: 503      1753 5
: 504      1754 5                  END
: 505      1755 4                  ELSE
: 506      1756 3                      BEGIN
: 507      1757 4
: 508      1758 4      |+_The buffer is an I/O buffer. The low seven bits of byte 11 are the
: 509      1759 4      |_-channel number.
: 510      1760 4
: 511      1761 4
: 512      1762 4          LOCAL
: 513      1763 4              CHAN;
: 514      1764 4
: 515      1765 4              CHAN = (.FIRQB [12, B_] AND 127);
: 516      1766 4
: 517      1767 4              IF (.CHAN LEQ 0) THEN BASS$STOP_IO (BASS$ILLIO_CHA);
: 518      1768 4
: 519      1769 4              BASS$CB_PUSH (.CHAN, LUB$K_LUN_MIN);
: 520      1770 4
: 521      1771 4              IF ( NOT .CCB [LUB$V_OPENED]) THEN BASS$STOP_IO (BASS$IO_CHANOT);
: 522      1772 4
: 523      1773 4              IF ((.FIRQB [16, W_] + .FIRQB [14, W_]) GTRU .CCB [LUB$W_RBUF_SIZE])
: 524      1774 5                  THEN
: 525      1775 4                      BASS$STOP_IO (BASS$ILLBYTCOU);
: 526      1776 4
: 527      1777 4              IF (.FIRQB [14, W_] EQL 0) THEN BASS$STOP_IO (BASS$NO_ROOUSE);
: 528      1778 4
: 529      1779 4              BUFADR = .CCB [LUB$A_RBUF_ADR] + .FIRQB [16, W_];
: 530      1780 4              BUFLN = .FIRQB [14, W_];
: 531      1781 4              END;
: 532      1782 3
: 533      1783 3
: 534      1784 3      |+_Copy the user's parameter string to the FIRQB.
: 535      1785 3      |_-
: 536      1786 3
```

```
537      CH$MOVE (20, FIRQB [22, B_], FIRQB [12, B_]);
538
539      !+ Now do the RSTS/E .MESAG function
540      !-
541      BPAS$MESAG (FIRQB, .BUFLEN, .BUFADR, BYTXFR);
542
543      !+ Put things back.
544      !-
545      CH$MOVE (20, FIRQB [12, B_], FIRQB [22, B_]);
546      FIRQB [14, W_] = .BYTXFR;
547
548      IF (.CCB NEQA 0) THEN BASS$CB_POP ();
549
550      STR$COPY_R (RET_STRING, %REF (40), .FIRQB_DESC [DSC$A_POINTER]);
551      END;
552
553      [FUN$K_UUFSS1, FUN$K_UUFSS2] :          ! File string scan
554      BEGIN
555
556      LOCAL
557          STATUS,
558          FSB : $FSB_DEF;
559
560      STATUS = BPAS$FSS (FIRQB, FSB, .CODE_STR [DSC$A_POINTER] + 2, .CODE_STR [DSC$W_LENGTH] - 2);
561
562      IF ( NOT .STATUS)
563      THEN
564          LIB$STOP (.STATUS)
565      ELSE
566          BEGIN
567              LOCAL
568                  STATUS;
569
570              STATUS = BPAS$FREE_BLOCK (.FSB [FSB$A_FSA], NAM$C_MAXRSS);
571
572              IF ( NOT .STATUS) THEN LIB$STOP (.STATUS);
573
574      !+ Return information to the user from the FIRQB and FSB.
575      !-
576      FIRQB [3, B_] = .FIRQB [5, B_];
577      FIRQB [16, W_] = .FIRQB [28, W_];
578      FIRQB [20, W_] = .FIRQB [30, W_];
579      FIRQB [28, W_] = .FSB [FSB$W_FLAG_1];
580      FIRQB [30, W_] = .FSB [FSB$W_FLAG_2];
581      STR$COPY_R (RET_STRING, %REF (30), .FIRQB_DESC [DSC$A_POINTER]);
582      END;
583
584      END;
585
586      [FUN$K_UUPRI] :          ! Set priority, etc.
587
588      !+ Only priority setting is implemented; all else is ignored.
589      !-
590      BEGIN
591          BPAS$SET_PRI (FIRQB);
592
593
```



```

: 594      1844 2      END;
: 595      1845 2
: 596      1846 2      [FUN$K_UUATR] :      ! Read/write file
: 597      1847 2      BAS$$STOP (BAS$K_MISSPEFEA);
: 598      1848 2
: 599      1849 2      [FUN$K_UUASS] :      ! Assign
: 600      1850 2      BEGIN
: 601      1851 2      BPAS$ASSIGN (FIRQB);
: 602      1852 2      STR$COPY_DX (RET_STRING, FIRQB_DESC);
: 603      1853 2      END;
: 604      1854 2
: 605      1855 2      [FUN$K_UUDEA] :      ! Deassign
: 606      1856 2      BEGIN
: 607      1857 2      BPAS$DEASSIGN (FIRQB);
: 608      1858 2      END;
: 609      1859 2
: 610      1860 2      [FUN$K_UUDAL] :      ! Deassign all
: 611      1861 2      BPAS$DEASS_ALL ();
: 612      1862 2
: 613      1863 2      [FUN$K_UUSDEF] :      ! Set default
: 614      1864 2      BEGIN
: 615      1865 2      BPAS$SET_DEF (FIRQB);
: 616      1866 2      END;
: 617      1867 2
: 618      1868 2      [INRANGE] :      ! Unimplemented
: 619      1869 2      BAS$$STOP (BAS$K_MISSPEFEA);
: 620      1870 2
: 621      1871 2      [OUTRANGE] :      ! Not defined
: 622      1872 2      BAS$$STOP (BAS$K_ILLSYSUSA);
: 623      1873 2      TES;
: 624      1874 2
: 625      1875 2      STR$COPY_DX (.RESULT_STR, RET_STRING);
: 626      1876 2      STR$FREE_T_DX (RET_STRING);
: 627      1877 2      RETURN;
: 628      1878 1      END;      ! of routine BAS$$UUO
```

00 00 00138 P.AAA: .BYTE 0, 0

.EXTRN R50ASC

OFFC 00000 BAS\$\$UUO:

			5A	00000000G	00	9E	00002	.WORD	Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11	: 1559	
			59	00000000G	00	9E	00009	MOVAB	BPAS\$MESAG, R10		
			5E	FD7C	CE	9E	00010	MOVAB	BAS\$\$STOP_10, R9		
48			AE	01020229	8F	D0	00015	MOVAB	-644(SP), -SP	1607	
4C			AE	52	AE	9E	0001D	MOVL	#16908841, FIRQB_DESC	1610	
			56	08	AC	D0	00022	MOVAB	FIRQB+2, FIRQB_DESC+4	1611	
0229	8F		04	B6	66	2C	00026	MOVL	CODE_STR, R6		
				52	AE		0002E	MOVCS	(R6), @4(R6), #0, #553, FIRQB+2		
			F8	AD	02000000	8F	D0	00030	MOVL	#33554432, RET_STRING	1612
				FC	AD	D4	00038	CLRL	RET_STRING+4	1615	
			E6	8F	53	AE	8F	0003B	CASEB	FIRQB+3, #-26, #54	1620
0214		36									
02C9		02C9		02C9	02C9		00041	1\$: .WORD	32\$-1\$,-		
		02C9		02C9	02C9		00049		32\$-1\$,-		

Page 16
(4)

32\$-1\$, -
21\$-1\$, -
32\$-1\$, -
32\$-1\$, -
32\$-1\$, -
32\$-1\$, -
32\$-1\$, -
32\$-1\$, -
32\$-1\$, -
32\$-1\$, -
26\$-1\$, -
32\$-1\$, -
32\$-1\$, -
21\$-1\$, -
32\$-1\$, -
32\$-1\$, -
2\$-1\$, -
32\$-1\$, -
32\$-1\$, -
32\$-1\$, -
32\$-1\$, -
32\$-1\$, -
32\$-1\$, -
32\$-1\$, -
32\$-1\$, -
32\$-1\$, -
32\$-1\$, -
32\$-1\$, -
32\$-1\$, -
32\$-1\$, -
32\$-1\$, -
38\$-1\$, -
27\$-1\$, -
28\$-1\$, -
30\$-1\$, -
32\$-1\$, -
32\$-1\$, -
32\$-1\$, -
32\$-1\$, -
32\$-1\$, -
5\$-1\$, -
32\$-1\$, -
32\$-1\$, -
32\$-1\$, -
12\$-1\$, -
32\$-1\$, -
32\$-1\$, -
32\$-1\$, -
32\$-1\$, -
32\$-1\$, -
31\$-1\$, -
#BASSK_I
33\$
#0, BASS
4\$

MOVZBL 315-15
BRW #BAS\$K-ILSYSUSA, -(SP)
CALLS 33\$
BRB #0, BAS\$CTRLC
4\$

1872
1624

40	AE	020E0700	8F	DO	000BF	3\$:	MOVL	#34471936, ERR_STRING	1633	
		44	AE	D4	000C7		CLRL	ERR_STRING+4	1636	
	7E	54	AE	9A	000CA		MOVZBL	FIRQB+4, -(SP)	1637	
		44	AE	9F	000CE		PUSHAB	ERR_STRING		
00000000G	00		02	FB	000D1		CALLS	#2, BASSERT		
38	AE	01020002	8F	DO	000D8		MOVL	#16908290, HEADER	1638	
3C	AE	FF1A	CF	9E	000E0		MOVAB	P.AAA, HEADER+4	1641	
		40	AE	9F	000E6		PUSHAB	ERR_STRING	1642	
		3C	AE	9F	000E9		PUSHAB	HEADER		
		F8	AD	9F	000EC		PUSHAB	RET_STRING		
00000000G	00		03	FB	000EF		CALLS	#3, STR\$CONCAT		
		40	AE	9F	000F6		PUSHAB	ERR_STRING	1643	
00000000G	00		01	FB	000F9		CALLS	#1, STR\$FREE1_DX		
		0212	31	00100	4\$:	BRW	34\$		1620	
		40	AE	9F	00103	5\$:	PUSHAB	ASCII_LOGNAM	1664	
		5A	AE	9F	00106		PUSHAB	FIRQB+6		
08	AE		06	DO	00109		MOVL	#6, 8(SP)		
		08	AE	9F	0010D		PUSHAB	8(SP)		
00000000G	00		03	FB	00110		CALLS	#3, R50ASC		
	3F	40	AE	91	00117		CMPB	ASCII_LOGNAM, #63	1666	
			07	12	0011B		BNEQ	6\$		
	7E	00G	8F	9A	0011D		MOVZBL	#BASSK_ILLSYSUSA, -(SP)		
	69		01	FB	00121		CALLS	#1, BASS\$STOP_10		
0042	03	FF	54	AE	8F	6\$:	CASEB	FIRQB+4, #-1, #3	1672	
001C	006C		0008		0012A	7\$:	.WORD	8\$-7\$, -		
								11\$-7\$, -		
								9\$-7\$, -		
								10\$-7\$, -		
OC	AE	5A	AE	14	28	00132	8\$:	MOV C3	#20, FIRQB+10, MESAG	1677
5C	AE	OC	AE	14	28	00138		MOV C3	#20, MESAG, FIRQB+12	1678
56	AE	40	AE	06	28	0013E		MOV C3	#6, ASCII_LOGNAM, FIRQB+6	1679
				50	11	00144		BRB	11\$	1672
56	AE	40	AE	06	28	00146	9\$:	MOV C3	#6, ASCII_LOGNAM, FIRQB+6	1687
		5E	AE	01	AE	0014C		MNEG W	#1, FIRQB+14	1688
		5D	AE	01	90	00150		MOV B	#1, FIRQB+13	1689
			04	AE	9F	00154		PUSHAB	BYTXFR	1690
				7E	7C	00157		CLRQ	-(SP)	
			5C	AE	9F	00159		PUSHAB	FIRQB	
		6A		04	FB	0015C		CALLS	#4, BP\$MESAG	
20	00	6E		00	2C	0015F		MOV C5	#0, (SP), #0, #32, FIRQB+2	1691
			52	AE		00164				
		54	AE	02	90	00166		MOV B	#2, FIRQB+4	1692
				2A	11	0016A		BRB	11\$	1672
56	AE	40	AE	06	28	0016C	10\$:	MOV C3	#6, ASCII_LOGNAM, FIRQB+6	1697
		54	AE	01	90	00172		MOV B	#1, FIRQB+4	1698
		5E	AE	01	AE	00176		MNEG W	#1, FIRQB+14	1699
		5D	AE	01	90	0017A		MOV B	#1, FIRQB+13	1700
			04	AE	9F	0017E		PUSHAB	BYTXFR	1701
				7E	7C	00181		CLRQ	-(SP)	
			5C	AE	9F	00183		PUSHAB	FIRQB	
		6A		04	FB	00186		CALLS	#4, BP\$MESAG	
20	00	6E		00	2C	00189		MOV C5	#0, (SP), #0, #32, FIRQB+2	1702
			52	AE		0018E				
		54	AE	0102	8F	B0	00190	MOV W	#258, FIRQB+4	1703
			04	AE	9F	00196	11\$:	PUSHAB	BYTXFR	1711
				7E	7C	00199		CLRQ	-(SP)	
			5C	AE	9F	0019B		PUSHAB	FIRQB	

1E	00	6A	04	FB	0019E	CALLS	#4, BP\$MESAG	
		6E	00	2C	001A1	MOV C5	#0, (SP), #0, #30, RES_STRING	1712
	28	AE	AE		001A6			
		5C	14	28	001A8	MOV C3	#20, FIRQB+12, RES_STRING+8	1713
			AE	9F	001AE	PUSHAB	RES_STRING	1714
			00FD	31	001B1	BRW	24\$	
			5C	AE	95 001B4	TSTB	FIRQB+12	1733
				17	12 001B7	BNEQ	14\$	
				5B	D4 001B9	CLRL	CCB	1739
		28	66	B1	001BB	CMPW	(R6), #40	1741
			0C	1B	001BE	BLEQU	13\$	
		57	7A	AE	9E 001C0	MOVAB	FIRQB+42, BUFADR	1744
		58	66	3C	001C4	MOVZWL	(R6), BUFLN	1745
		58	28	C2	001C7	SUBL2	#40, BUFLN	
			58	11	001CA	BRB	19\$	1741
			57	7C	001CC	CLRQ	BUFADR	1752
			54	11	001CE	BRB	19\$	1741
		07	00	EF	001D0	EXTZV	#0, #7, FIRQB+12, CHAN	1766
		7E	07	14	001D6	BGTR	15\$	1768
		69	00G	8F	9A 001D8	MOVZBL	#BAS\$K_ILLIO_CHA, -(SP)	
				01	FB 001DC	CALLS	#1, BAS\$\$STOP_IO	
				50	D4 001DF	CLRL	R0	1770
			00000000G	00	16 001E1	JSB	BAS\$\$CB_PUSH	
		07	FC	AB	E8 001E7	BLBS	-4(CCB), 16\$	1772
		7E	00G	8F	9A 001EB	MOVZBL	#BAS\$K_IO_CHANOT, -(SP)	
		69		01	FB 001EF	CALLS	#1, BAS\$\$STOP_IO	
		50	60	AE	3C 001F2	MOVZWL	FIRQB+16, R0	1774
		51	5E	AE	3C 001F6	MOVZWL	FIRQB+14, R1	
		50		51	C0 001FA	ADDL2	R1, R0	
50	D2	AB	10	00	ED 001FD	CMPZV	#0, #16, -46(CCB), R0	
			07	1E	00203	BGEQU	17\$	
		7E	00G	8F	9A 00205	MOVZBL	#BAS\$K_ILLBYTCOU, -(SP)	1776
		69		01	FB 00209	CALLS	#1, BAS\$\$STOP_IO	
			5E	AE	B5 0020C	TSTW	FIRQB+14	1778
				07	12 0020F	BNEQ	18\$	
		7E	00G	8F	9A 00211	MOVZBL	#BAS\$K_NO_R0OUSE, -(SP)	
		69		01	FB 00215	CALLS	#1, BAS\$\$STOP_IO	
		57	60	AE	3C 00218	MOVZWL	FIRQB+16, BUFADR	1780
		57	EC	AB	C0 0021C	ADDL2	-20(CCB), BUFADR	
		58	5E	AE	3C 00220	MOVZWL	FIRQB+14, BUFLN	1781
	5C	AE	66	AE	14 28 00224	MOV C3	#20, FIRQB+22, FIRQB+12	1787
				08	AE 9F 0022A	PUSHAB	BYTXFR	1791
				57	DD 0022D	PUSHL	BUFADR	
				58	DD 0022F	PUSHL	BUFLN	
			5C	AE	9F 00231	PUSHAB	FIRQB	
		6A		04	FB 00234	CALLS	#4, BP\$MESAG	
	66	AE	5C	AE	14 28 00237	MOV C3	#20, FIRQB+12, FIRQB+22	1795
		5E	AE	08	AE B0 0023D	MOVW	BYTXFR, FIRQB+14	1796
				5B	D5 00242	TSTL	CCB	1798
				06	13 00244	BEQL	20\$	
			00000000G	00	16 00246	JSB	BAS\$\$CB_POP	
			4C	AE	DD 0024C	PUSHL	FIRQB_DESC+4	1800
		04	AE	28	D0 0024F	MOVL	#40, 4(SP)	
				60	11 00253	BRB	25\$	
		7E		66	3C 00255	MOVZWL	(R6), -(SP)	1810
		6E		02	C2 00258	SUBL2	#2, (SP)	
	7E	04	A6	02	C1 0025B	ADDL3	#2, 4(R6), -(SP)	

		2C	AE	9F	00260	PUSHAB	FSB	:
		5C	AE	9F	00263	PUSHAB	FIRQB	:
00000000G	00		04	FB	00266	CALLS	#4, BPAS\$FSS	1812
	0B		50	E8	0026D	BLBS	STATUS, 22\$	1814
00000000G	00		50	DD	00270	PUSHL	STATUS	:
			01	FB	00272	CALLS	#1, LIB\$STOP	:
	7E	FF	78	11	00279	BRB	29\$:
			8F	9A	0027B	MOVZBL	#255, -(SP)	1821
		2C	AE	DD	0027F	PUSHL	FSB+4	:
00000000G	00		02	FB	00282	CALLS	#2, BPAS\$FREE_BLOCK	:
	09		50	E8	00289	BLBS	STATUS, 23\$	1823
			50	DD	0028C	PUSHL	STATUS	:
00000000G	00		01	FB	0028E	CALLS	#1, LIB\$STOP	:
	53	55	AE	90	00295	MOVB	FIRQB+5, FIRQB+3	1828
	60	6C	AE	B0	0029A	MOVW	FIRQB+28, FIRQB+16	1829
	64	6E	AE	B0	0029F	MOVW	FIRQB+30, FIRQB+20	1830
	6C	46	AE	B0	002A4	MOVW	FSB+34, FIRQB+28	1831
	6E	44	AE	B0	002A9	MOVW	FSB+32, FIRQB+30	1832
		4C	AE	DD	002AE	PUSHL	FIRQB_DESC+4	1833
	04		1E	DD	002B1	MOVL	#30, 4(SP)	:
		04	AE	9F	002B5	PUSHAB	4(SP)	:
		F8	AD	9F	002B8	PUSHAB	RET_STRING	:
00000000G	00		03	FB	002BB	CALLS	#3, STR\$COPY_R	:
			51	11	002C2	BRB	34\$	1620
		50	AE	9F	002C4	PUSHAB	FIRQB	1843
00000000G	00		01	FB	002C7	CALLS	#1, BPAS\$SET_PRI	:
			45	11	002CE	BRB	34\$	1620
		50	AE	9F	002D0	PUSHAB	FIRQB	1851
00000000G	00		01	FB	002D3	CALLS	#1, BPAS\$ASSIGN	:
		48	AE	9F	002DA	PUSHAB	FIRQB_DESC	1852
		F8	AD	9F	002DD	PUSHAB	RET_STRING	:
00000000G	00		02	FB	002E0	CALLS	#2, STR\$COPY_DX	:
			2C	11	002E7	BRB	34\$	1620
		50	AE	9F	002E9	PUSHAB	FIRQB	1857
00000000G	00		01	FB	002EC	CALLS	#1, BPAS\$DEASSIGN	:
			20	11	002F3	BRB	34\$	1620
00000000G	00		00	FB	002F5	CALLS	#0, BPAS\$DEASS_ALL	1861
			17	11	002FC	BRB	34\$:
		50	AE	9F	002FE	PUSHAB	FIRQB	1865
00000000G	00		01	FB	00301	CALLS	#1, BPAS\$SET_DEF	:
			0B	11	00308	BRB	34\$	1620
	7E	00G	8F	9A	0030A	MOVZBL	#BAS\$K MISSPEFEA, -(SP)	1869
00000000G	00		01	FB	0030E	CALLS	#1, BAS\$\$\$STOP	:
		F8	AD	9F	00315	PUSHAB	RET_STRING	1875
		04	AC	DD	00318	PUSHL	RESULT_STR	:
00000000G	00		02	FB	0031B	CALLS	#2, STR\$COPY_DX	:
		F8	AD	9F	00322	PUSHAB	RET_STRING	1876
00000000G	00		01	FB	00325	CALLS	#1, STR\$FREE1_DX	:
			04	0032C	RET			1878

; Routine Size: 813 bytes, Routine Base: _BAS\$CODE + 013A

: 629 1879 1 END
: 630 1880 1
: 631 1881 0 ELUDOM

! of module BAS\$SYS

PSECT SUMMARY

```

:
:      Name                Bytes                Attributes
:
:  _BASSCODE              1127 NOVEC,NOWRT, RD ,  EXE,  SHR,  LCL,  REL,  CON,  PIC,ALIGN(2)

```

Library Statistics

```

:
:      File                ----- Symbols ----- Pages Processing
:                        Total   Loaded   Percent   Mapped   Time
:
:  _$255$DUA28:[SYSLIB]STARLET.L32;1    9776        13         0       581     00:01.2

```

COMMAND QUALIFIERS

```

:
:  BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/NOTRACE/LIS=LIS$:BASSYS/OBJ=OBJ$:BASSYS MSRC$:BASSYS/UPDATE=(ENH$:BASSYS)
:
:  Size:                1125 code + 2 data bytes
:  Run Time:            00:25.7
:  Elapsed Time:        00:58.6
:  Lines/CPU Min:       4388
:  Lexemes/CPU-Min:     31558
:  Memory Used:         288 pages
:  Compilation Complete

```


0032 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

